93-02358/

National Defence / Défense nationale

SIMULATION OF DOWNLINK SYNCHRONIZATION FOR A FREQUENCY-HOPPED SATELLITE COMMUNICATION SYSTEM (U)

by

Lyle Wagner
Communications and Navigation Section
Electronics Division

DEFENCE RESEARCH ESTABLISHMENT OTTAWA
TECHNICAL NOTE 92-9

PCN
D6470

April 1992
Ottawa

ABSTRACT

Many military communication systems are required to operate in environments where an enemy may attempt to interfere with the communication. One of the most serious threats is the use of jammers. This threat has caused designers of communications systems to counter with the use of spread spectrum techniques such as frequency-hopping.

Frequency-hopped spread spectrum satellite communication systems provide robust communications resilient to many types and levels of jamming. Unfortunately this increase in resilience is offset by an increase in complexity while establishing the communication link, termed synchronization. This document describes a downlink synchronization process that involves the transmission of synchronization hops by the satellite and a two-step ground terminal synchonization procedure. In addition computer simulation results are documented which demonstrate that the proposed synchronization algorithm is able to operate in environments as severe as the communication link.

RÉSUMÉ

Beaucoup de systèmes militaires de communication sont requis pour opérer dans des milieux où un ennemi peut essayer d'interférer avec les communications. Une des menaces les plus sérieuses est l'utilisation de brouilleurs. Cette menace a forcé la défense des systèmes de communication en utilisant des techniques d'étalement du spectre telles que le saut en fréquence.

Les systèmes de communication par satellite avec étalement du spectre par saut de fréquence fournissent des communications robustes, protégées contre plusieurs types et niveaux de brouillage. Malheureusement, cette protection additionnelle est aux dépens d'une complexité accrue de la synchronisation, laquelle marque le début du lien de communication. Ce document décrit un processus de synchronisation du lien descendant impliquant la transmission de sauts de synchronisation par le satellite et une procédure de synchronisation à deux étapes pour la station terrestre. De plus, des résultats de simulation par ordinateur sont inclus, lesquels démontrent que l'algorithme proposé de synchronisation est capable d'opérer dans des conditions similaires à celles anticipées pour le lien de communication.

EXECUTIVE SUMMARY

Many military communication systems are required to operate in environments where an enemy may attempt to interfere with the communication. One of the most serious threats is the use of jammers. This threat has caused designers of communications systems to counter with the use of spread spectrum techniques such as frequency-hopping.

Frequency-hopped spread spectrum satellite communication systems provide robust communications resilient to many types and levels of jamming. Unfortunately this increase in resilience is offset by an increase in complexity while establishing the communication link, termed synchronization. In order for the entire communication system to be effective against hostile and inadvertent interference the communication link and synchronization process must be protected to the same level.

This document examines a two-step downlink synchronization method. The method involves the transmission by the satellite of synchronization (sync) hops that are modulated with a synchronization code and are received by the ground terminal where the two-step algorithm is implemented. The first step is the envelope detection of the sync hops, coarse time synchronization. When the terminal is confident that the coarse synchronization process is complete the second step is activated, fine time synchronization. Fine time synchronization involves correlating a downlink sync hop with an internally generated reference sync hop to obtain an accurate time estimate. Additionally, a prediction of the frequency offset can be made to fine tune the local oscillator.

Computer simulations were made of the synchronization process using synchronization codes of varying lengths to verify the algorithm. These simulations demonstrate that the proposed algorithm is capable of operating under severe jamming conditions and is compatible with the corresponding communication link.

TABLE OF CONTENTS

Abstract iii

Résumé iii

Executive Summary v

Table of Contents vi

List of Figures ix

Acknowledgement xi

1. Introduction 1

2. Synchronization Process 2

3. Computer Simulation 13

4. Simulation Results 15

5. Conclusion 27

References 28

LIST OF FIGURES

1A	Autocorrelation plot for 13-bit code.	3
1B	Autocorrelation plot for 28-bit code.	3
1C	Autocorrelation plot for 112-bit code.	3
2	Structure of synchronization burst.	4
3	Block diagram of sync hop detection and coarse time estimate.	6
4	Autocorrelation triuangle of the sync envelope.	7
5	Block diagram of fine time estimate.	10
6	Coarse timing accuracy for a 13-bit synchronization code.	16
7	Corase timing accuracy for a 28-bit synchronization code.	16
8	Coarse timing accuracy for a 112-bit synchronization code.	16
9	Fine timing estimate for a 13-bit sync code using one hop.	17
10	Fine timing estimate for a 28-bit sync code using one hop.	17
11	Fine timing estimate for a 112-bit sync code using one hop.	17
12	Fine timing estimate for a 13-bit sync code using 4 hops.	18
13	Fine timing estimate for a 28-bit sync code using 4 hops.	18
14	Fine timing estimate for a 112-bit sync code using 4 hops.	18
15	Ratio of 4-hop fine timing estimate over 1-hop fine timing estimate.	20
16	Frequency offset calcualtion accuracy using one hop	21
17	Frequency offset calcualtion accuracy using 4 hops	21
18	Frequency offset determination accuarcy.	22
19	Ambiguity diagram for 13-bit sync code.	24
20	Ambiguity diagram for 28-bit sync code.	25
21	Ambiguity diagram for 112-bit sync code.	26

ACKNOWLEDGEMENT

The author would like to to thank Bill Seed for his help in explaining the theory in terms simple enough that even the author could understand. The author would also like to acknowledge the help of Aaron Gulliver and his software in determining the the 28-bit and 112-bit codes.

1.0

INTRODUCTION

Frequency-hopped (FH) spread spectrum satellite communication systems provide a robust communications link resilient to both hostile and inadvertent interference. Unfortunately the increase in resilient is offset by an increase in complexity of the initiation of the communication link, i.e. the synchronization of the ground terminal with the satellite. If the system is to perform satisfactorily in a hostile environment the synchronization process must be at least as robust as the communications link.

This document describes the simulation of a downlink synchronization process proposed for a FH communications system. An explanation of the synchronization algorithm is given in the section 2 followed by a description of the simulation process in section 3. The results of the simulation process are presented in section 4 with a conclusion of the results in section 5.

2.0

SYNCHRONIZATION PROCESS

This section describes the synchronization process used in this document. The downlink synchronization process involves spatial, temporal, and frequency synchronization. The spatial synchronization process is outside the scope of this document and is assumed to have been achieved before initiating temporal and frequency synchronization. The equations used in the determination of the coarse time estimate, fine time estimate, and frequency offset are derived. In addition lower bounds for the coarse and fine time estimate are developed.

The downlink synchronization process (for the system under discussion) is done within the ground terminal with the aid of synchronization (sync) hops transmitted by the satellite. The terminal starts by detecting the sync hops and calculating a coarse time estimate (coarse time synchronization), followed by calculating a fine time estimate (fine time synchronization). When downlink time synchronization is complete, an estimate of the frequency offset is made. The estimate of frequency offset can be used by the downlink receiver to fine tune the local oscillator.

The sync hops are modulated using phase-shift keying (PSK) with a code chosen for its good autocorrelation properties. That is, the code has the lowest maximum sidelobe for the given code length. Three different lengths were chosen: a short 13-bit Barker code, a medium length 28-bit code, and a long 112-bit code. The 28-bit and 112-bit codes were found by using a computer search technique to select the best code. The codes used in this document are given in Table 1. The autocorrelation plots for these codes are given in Figures 1A, 1B, and 1C.

Code length	Code sequence (hexadecimal)
13 bits	159F
28 bits	18FD 5B6
112 bits	FA74 2D84 1B42 E5F3 8C95 D5D4 98E7

Table 1 Synchronization codes for various code lengths.

Sync hops are transmitted by the satellite in bursts. A sync burst consists of 4 sync hops, each separated by one hop. The structure of a sync burst is shown in Figure 2.

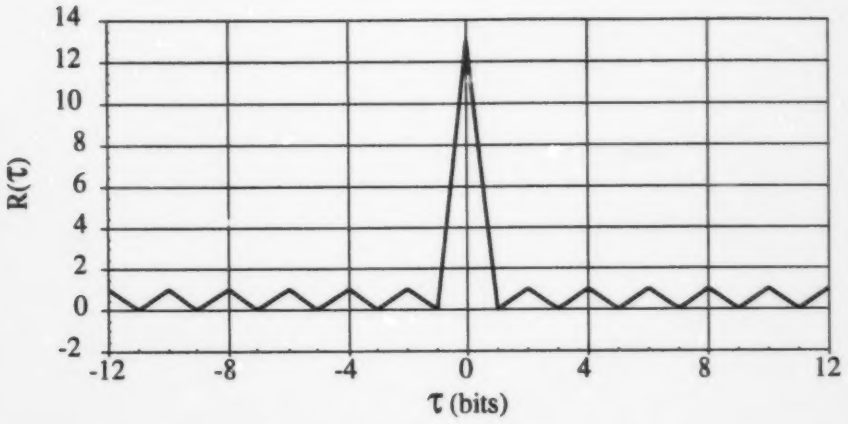

Figure 1A Autocorrelation plot for 13-bit code.

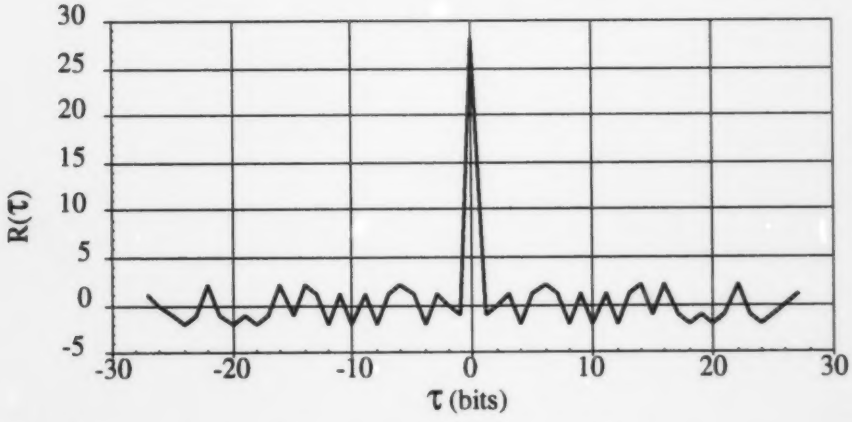

Figure 1B Autocorrelation plot for 28-bit code.

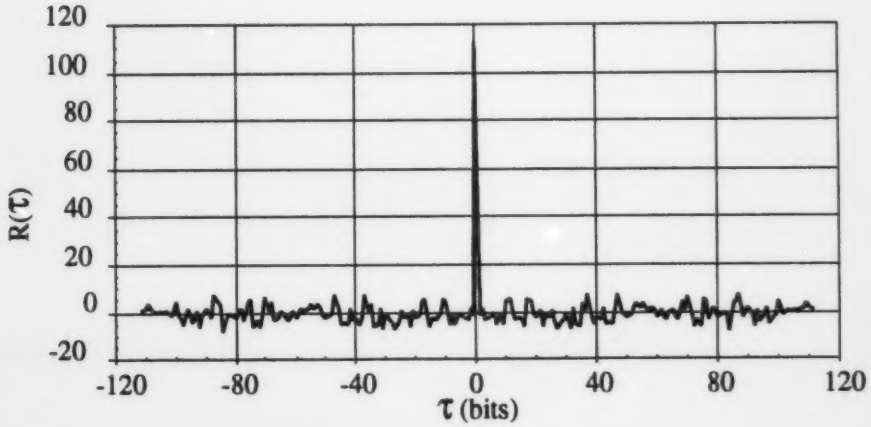

Figure 1C Autocorrelation plot for 112 bit-code.

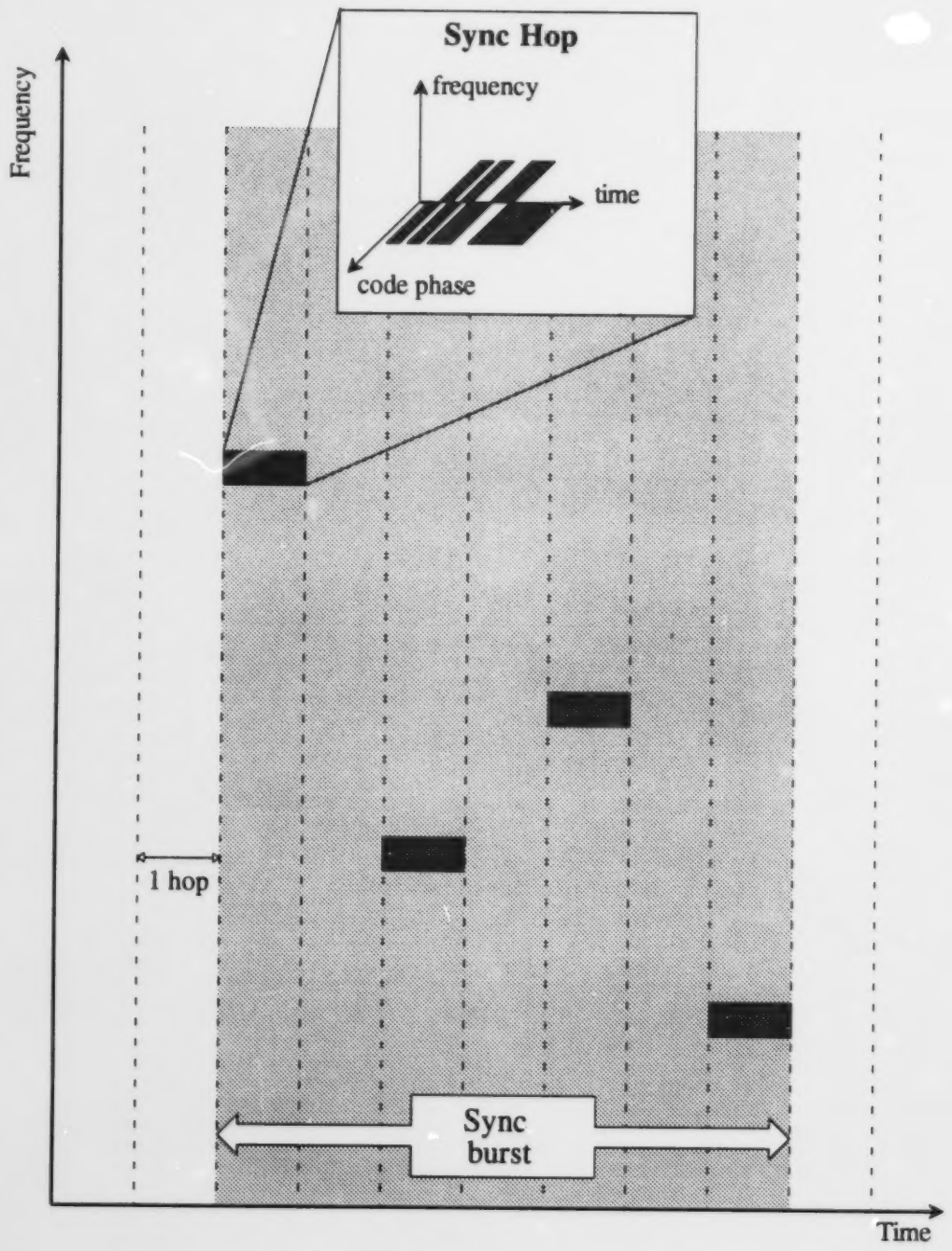

Figure 2 Structure of synchronization burst.

Sync Hop Detection Algorithm

The first step in achieving synchronization is to detect the sync burst (envelope detection). Referring to Figure 3, the detection process is:

a. the terminal dehopping synthesizer is set to the frequency of the first hop of the synchronization burst.

b. the downlink signal is dehopped, bandpass filtered, squared, and lowpass filtered.

c. the resultant analog signal is sampled (3 samples per hop) and converted to a digital value.

d. the digital sampled value is added to the two previous samples and compared to a predetermined threshold.

e. when the threshold is exceeded the synchronization process jumps ahead and looks for the next sync hop.

f. if the threshold is exceeded for three consecutive sync hops the sync hop burst is assumed to be detected and the fourth sync hop in the series is used to calculate the coarse timing estimate. If a sync hop is not detected in the second or third hop, the process resets (i.e. dehopping synthesizer is set to the frequency of the first hop) and continues the search.

Coarse Time Estimate Calculation

Once the sync burst is detected, the fourth hop of the burst is used to calculate the coarse time estimate. The output from the detection algorithm traces out the autocorrelation triangle of the sync hop envelope. The peak of this triangle indicates the point of zero timing offset. The base of the triangle is two hop intervals wide (see Figure 4). The timing mismatch from the largest autocorrelation is given by (1), which is derived from the geometry of the triangle [1, equation (6)].

$$t_{off} = \frac{V_2 t_{hop} - V_1(t_{hop} - t_s)}{V_1 + V_2} \qquad \text{.....(1)}$$

where
t_{hop} = width of hop
t_s = time between two samples
V_1 = amplitude of largest autocorrelation value
V_2 = amplitude of second largest autocorrelation value
t_1, t_2 = time of sample V_1, V_2

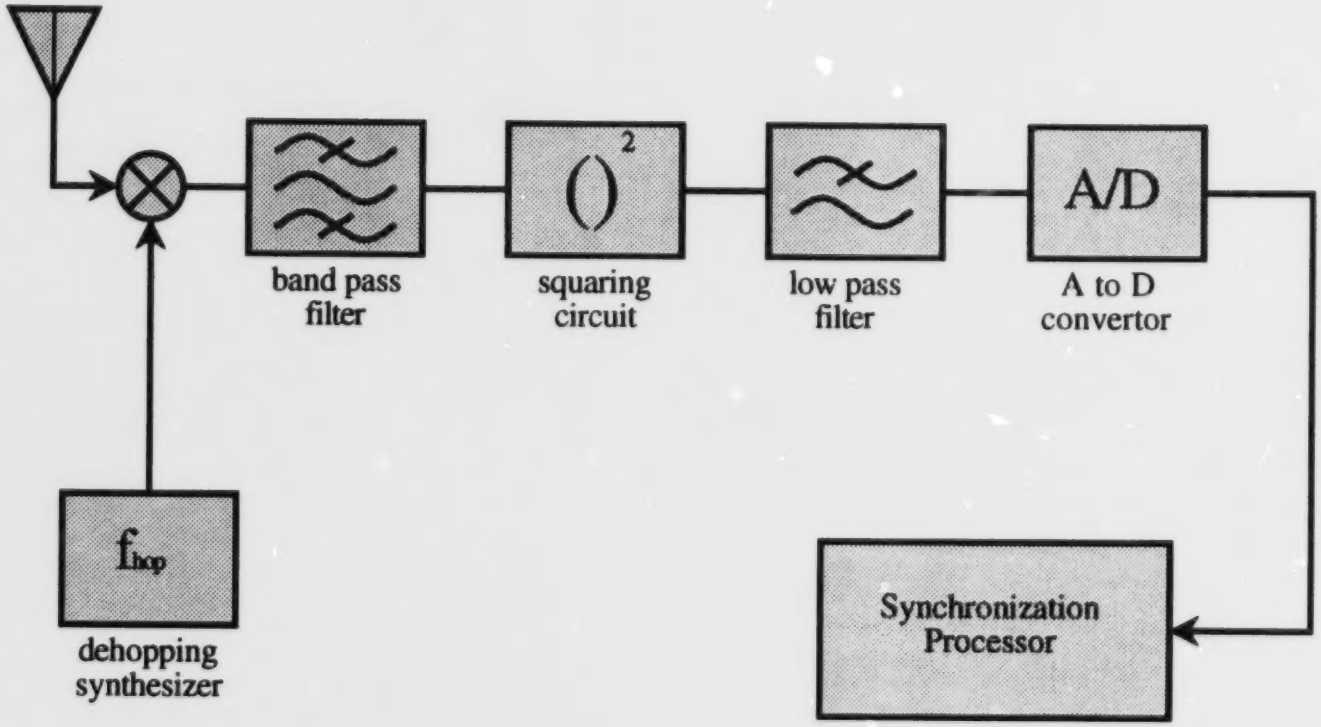

Figure 3 Block diagram of sync hop detection and coarse time estimate.

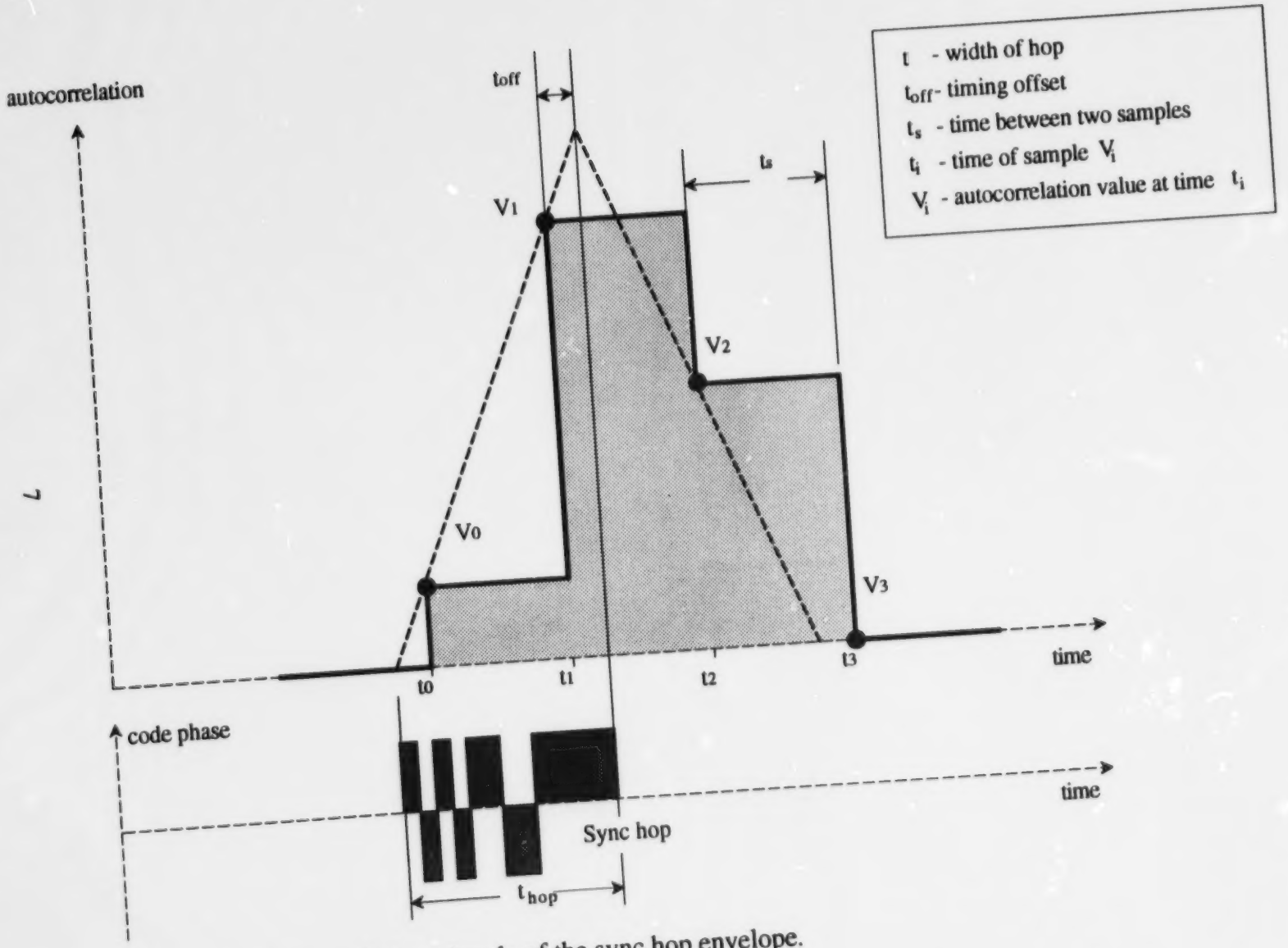

Figure 4 Autocorrelation triangle of the sync hop envelope.

Coarse timing uncertainty bound

Employing techniques first used in radar to determine the accuracy of radar measurements, a lower bound on the accuracy of the timing estimate can be derived. This bound is usually referred to as the Cramer-Rao (CR) bound. The general time-delay rms error lower bound (δT_R) is defined by [2, equation 11.17]

$$\delta T_R \geq \frac{1}{\beta\sqrt{\frac{2E}{N_0}}} \qquad \text{.....(2)}$$

where: β = effective bandwidth,
E = signal energy, and
N_0 = noise spectral density.

To derive the lower-bound for coarse for our simulation, we first make the assumption that the signal is a bandwidth-limited rectangular pulse [2, equation 11.18] to obtain

$$\beta^2 = \frac{1}{\tau^2}\,\frac{\pi B\tau - \sin\pi B\tau}{\mathrm{Si}(\pi B\tau) + (\cos\pi B\tau - 1)/\pi B\tau} \qquad \text{.....(3)}$$

where $\mathrm{Si}(\pi B\tau) = \int_0^{\pi B\tau} \frac{\sin(u)}{u}\,du$, the sine integral function

B = bandwidth of filter
τ = pulse width

The normalized timing uncertainty bound can de derived by dividing both sides of (3) by τ; replacing the energy E by NE_b (N is the number of bits in the hop and E_b is the energy of one bit); replacing τ, the hop duration, by $N\tau_b$ (τ_b is the duration of one bit); and re-arranging terms giving

$$\frac{\delta T_R}{\tau_b} = \frac{1}{(\beta\tau)\sqrt{\frac{2}{N}\frac{E_b}{N_0}}} \qquad \text{.....(4)}$$

where β is derived from (3).

Equation (5) is the normalized timing uncertainty bound for a continuous wave (CW) pulse. For our simulation, the E_b/N_0 ratio should be replaced by an effective E_b/N_0 which includes

losses due to PSK modulation. The SNR for a PSK modulated signal of the lowpass component of the signal is given in [4] as

$$SNR_{PSK} = \frac{N\rho_{PSK}^2}{(\xi + 2r_{PSK})} \qquad \text{.....(5)}$$

where: $\rho_{PSK} = E_b/N_0$ for a PSK modulated pulse
$\xi = 2$, the time bandwidth product

The SNR for an equivalent CW pulse with same ξ can be derived as

$$SNR_{CW} = \frac{N^2 \rho_{CW}^2}{(7/4 + 2N\rho_{CW})} \qquad \text{.....(6)}$$

where: $\rho_{CW} = E_b/N_0$ for a CW pulse

Using the same technique as [4], an expression can be derived that transforms the E_b/N_0 for a PSK modulated pulse to the effective E_b/N_0 for a CW pulse. This transform is given by

$$r_{CW} = \frac{\rho_{PSK}^2 + \rho_{PSK}^2\sqrt{1 + \frac{7}{4N}\left(\frac{2}{\rho_{PSK}^2} + \frac{2}{\rho_{PSK}}\right)}}{2+2\rho_{PSK}} \qquad \text{.....(7)}$$

Therefore the lower bound for normalized coarse timing uncertainty is:

$$\frac{\delta T_R}{\tau_b} \geq \frac{1}{(\beta\tau)\sqrt{\frac{2}{N}\rho_{CW}}} \qquad \text{.....(8)}$$

Fine Time Estimate Calculation

After a coarse time estimate has been obtained, fine time estimation is done on the next sync burst. Referring to Figure 5, the time estimation process is:

a. the terminal dehopping synthesizer is set to the frequency of the first sync hop in the next sync burst.

b. the downlink signal is dehopped and bandpass filtered.

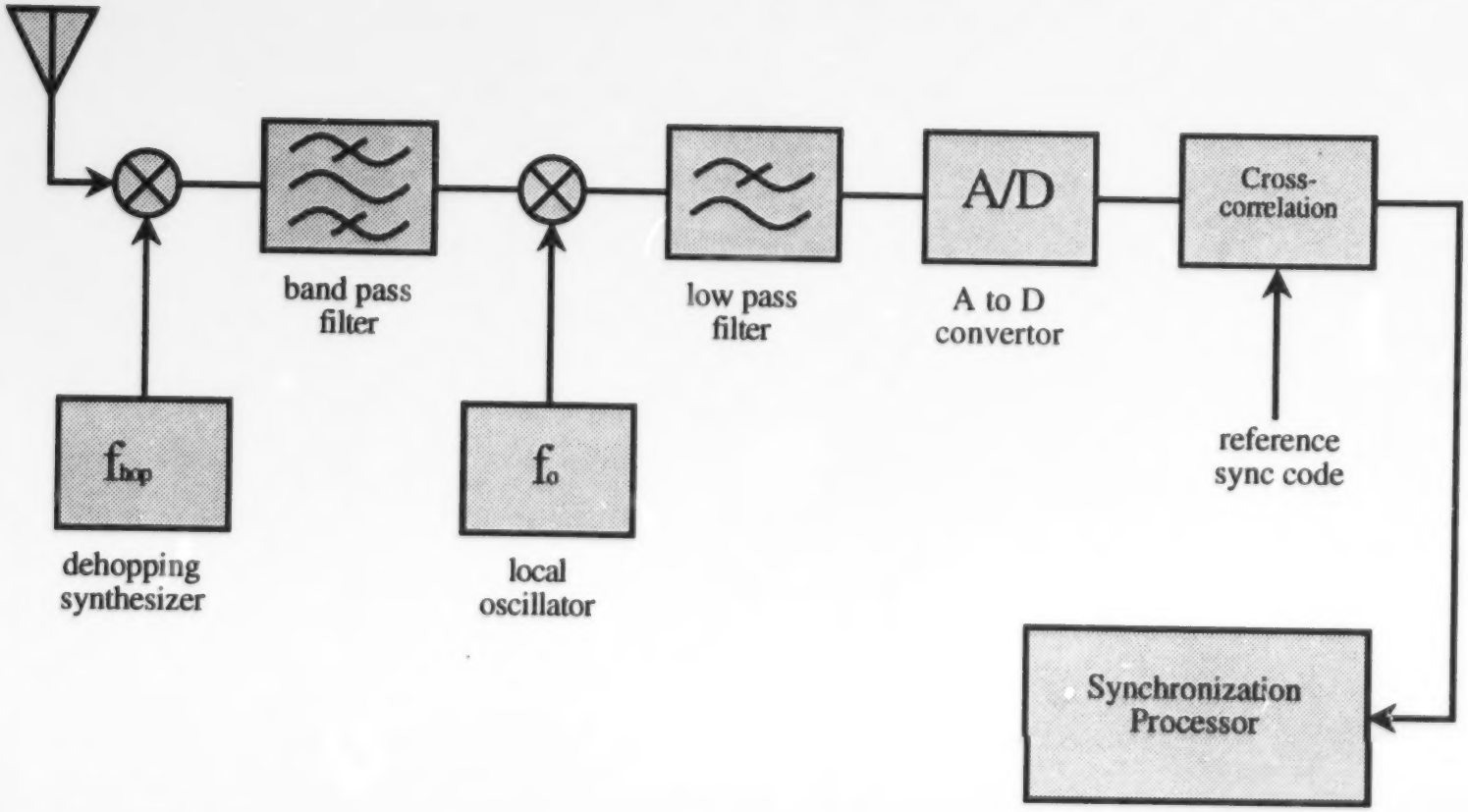

Figure 5 Block diagram of fine time estimate.

c. the hop is demodulated by mixing the signal with the expected offset frequency (f_o). The observation time starts 1/2 a hop before the coarse time estimate and extends 1/2 hop past the estimate stop time. This allows for a possible coarse time estimate error of ± 1/2 hop.

d. the resultant signal is sampled (2 samples per code bit).

e. the digital sample is cross-correlated with the sync code to obtain a matched filter output. As with the coarse timing, a triangle is traced with the peak indicating the point of zero timing mismatch with the base two bits wide.

Calculation of the timing offset is derived from the same geometry as (1) and is given by (2). The fine time estimate of t_o is given by

$$t_o = \frac{V_2 t_b - V_1(t_b - t_s)}{V_1 + V_2} \qquad \text{.....(9)}$$

where t_b = width of a bit
t_s = time between two samples
V_1 = amplitude of largest crosscorrelation value
V_2 = amplitude of second largest crosscorrelation value

Fine timing uncertainty bound

To derive the fine timing estimate lower bound, we make use of the CR bound and calculate the value of β from (3) using numerical integration of [2, equation 11.16]

$$\beta^2 = \frac{1}{\tau^2} \frac{\int_{\infty}^{-\infty} (2\pi f)^2 [S(f)]^2 \, df}{\int_{\infty}^{-\infty} [S(f)]^2 \, df} \qquad \text{.....(10)}$$

where: $S(f)$ = is the Fourier transform of the input signal s(t).

The value of β is dependent upon the bandwidth and shape of the input filter. The values of βτ (fine time synchronization) for the filters and pulse used in the simulations are given in Table 2. The lower bound for normalized fine timing uncertainty is:

$$\frac{\delta T_R}{t_b} \geq \frac{1}{\beta\tau\sqrt{\frac{2}{N}\frac{E_b}{N_0}}} \qquad \text{.....(11)}$$

Pulse duration t_b (hop)	Time bandwidth product	$\beta\tau$	Filter corner frequency (Hz)
1/13	1	1.882	6.5
1/13	2	2.201	13.0
1/13	4	2.979	26.0
1/13	6	3.685	39.0
1/28	2	2.149	28.0
1/28	4	2.979	56.0
1/28	6	4.262	112.0
1/56	2	2.139	56.0
1/56	4	2.971	56.0
1/112	2	2.135	112.0

Table 2 Values of $\beta\tau$ for various time bandwidth products (fine time sync).

Offset frequency calculation

An estimate of frequency error can be made by comparing the matched filter outputs of the sync hop at 0, f_s and $-f_s$, where f_s is chosen to be 30% of the hop rate for the simulations described in this document. The frequency estimate is approximated by

$$f_{est} = \frac{(P_{f_s} - P_{-f_s}) \times \Delta f \times f_{Correction}}{P_{f_s} + P_{-f_s} + P_0} \qquad \text{.....(12)}$$

where $P_k = |X_k|^2$

$$X_k = \int_{-\infty}^{\infty} x(t)\ e^{-j2\pi k}\, dt$$

x(t) = input signal with coding removed
$f_{Correction}$ = experimentally derived value to minimize estimate bias
k = $-f_s$, 0, or f_s

The normalized lower CR bound for the frequency estimation method is obtained from [1, equation 16] as

$$\frac{\delta f_o}{R_h} = \frac{0.433}{\sqrt{E_b/N_0}} \qquad \text{.....(13)}$$

3.0

COMPUTER SIMULATION

The synchronization process was simulated using a Monte Carlo method written in FORTRAN. The simulations were run on a VAX computer using the VMS operating system and on a Macintosh. The simulation software was divided into 3 main segments; transmitter, channel, and receiver. Each of these segments is described in the following paragraphs.

Transmitter

The transmit segment simulates the function of the payload in forming the sync hops and transmitting them to the user terminals. The sync hop waveform is modulated with a sync code using binary PSK (BPSK) signal modulation. The sync code is one of the three codes described in Table 1. The BPSK signal is then upconverted by mixing with the carrier frequency plus some optional frequency offset. The frequency offset represents a mix of oscillator offset and Doppler frequency offset calculation errors. The offset is a user selected parameter. The frequency hopping portion of the system is not simulated; however the initial phase of each "hop" is a uniform random variable from 0 to 2π .

The sync burst is a digital representation of Figure 2 and is 7 hops in length. The computer representation is a two dimensional array with one dimension being the amplitude of the digital signal and the other dimension being sample number. This sync burst is placed into a larger array with a random starting sample number. The random start time is the actual timing offset, t_o. This larger array represents the total time for a synchronization attempt (a synchronization attempt is defined in section 4).

Channel

The channel segment simulates the waveform being perturbed by suming additive white Gaussian noise (AWGN) to the output array from the transmit section of the program. The power of the signal is fixed while the power of the AWGN is variable. The ratio of the two determines the E_b/N_0 for the simulation. The AWGN is generated from normally distributed random numbers using the GASDEV function from [3].

Receiver

The receive segment simulates the downlink synchronization function of the ground terminal. The sync hop plus noise is first downconverted and then split into the coarse and fine synchronization branches. Each signal is then bandpass filtered with a user specified infinite impulse response (IIR) filter.

The thresholds used for coarse synchronization on the first 3 hops are user selectable and for this report are identical. The number of coarse samples per hop, the number of fine samples per bit, and the number of crosscorrelations values used in the fine time estimate are also user selectable. The values for the simulation results described in section 4 are:

Number of coarse samples per hop	3
Number of fine samples per hop	2
Number of matched filter values per estimate	1

To increase the accuracy of the estimate of the timing offset, t_o, the number of fine or coarse samples per bit could be increased. This would have the effect of more accurately tracing the crosscorrelation peak.

4.0

SIMULATION RESULTS

The results of the computer simulations of the synchronization process are described in this section. Each computer run consisted of a user specified number of synchronization attempts. An attempt is classified as successful (sync burst found), false alarm (software not synchronized to a sync burst), or unsuccessful (synchronization was not obtained in 10 sync hop bursts). As expected the number of false alarms and unsuccessful attempts increased as the E_b/N_0 decreased. To insure the statistical validity of each computer run a minimum of 500 successful synchronizations were collected.

Coarse Time Estimates

Simulations to determine coarse time estimates were done for three code lengths: 13 bits, 28 bits, and 112 bits. The RMS timing uncertainty (in bits) from the simulations are compared to the CR bound , refer to (7). The results are shown in Figures 6, 7, and 8 versus E_b/N_0. The agreement between the simulation and bound for high E_b/N_0 is very good. However for lower values of E_b/N_0 the agreement deteriates. This is to be expected because of the assumption of large E_b/N_0 in the derivation of the lower bound.

One important result from the simulation is that the synchronization algorithm was able to function at low values of E_b/N_0. In all cases the algorithm was able function down to an $E_b/N_0$of 15 dB. However, at times the algorithm had to cycle many times before a coarse estimate was obtained.

Fine Time Estimates

Simulations to determine fine time estimates were done for the same three code lengths (13 bits, 28 bits, and 112 bits) as for coarse time simulations. A filter with a time-bandwidth product of 2 was used. The RMS timing uncertainty (in bits) from the simulations compared to the CR bound are given in Figures 9, 10, and 11 for estimates based on one hop. The RMS error for timing uncertainty estimates based on a four hop average are shown in Figures 12, 13, and 14.

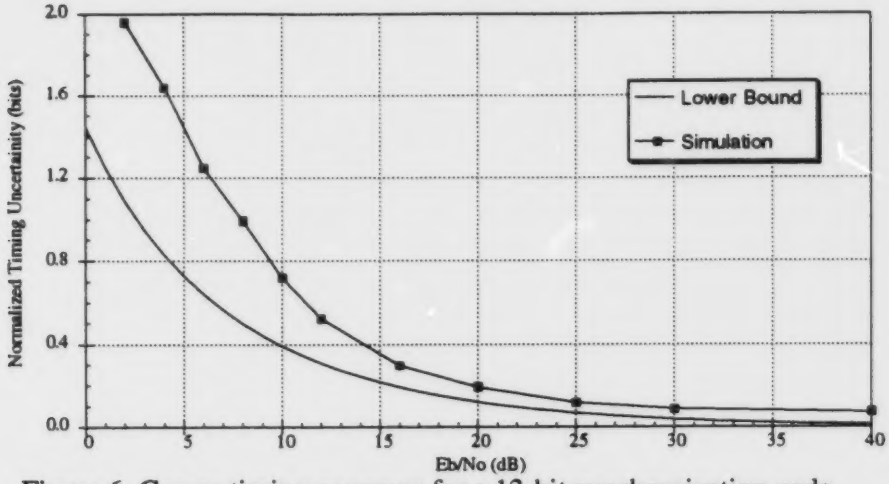

Figure 6 Coarse timing accuracy for a 13-bit synchronization code.

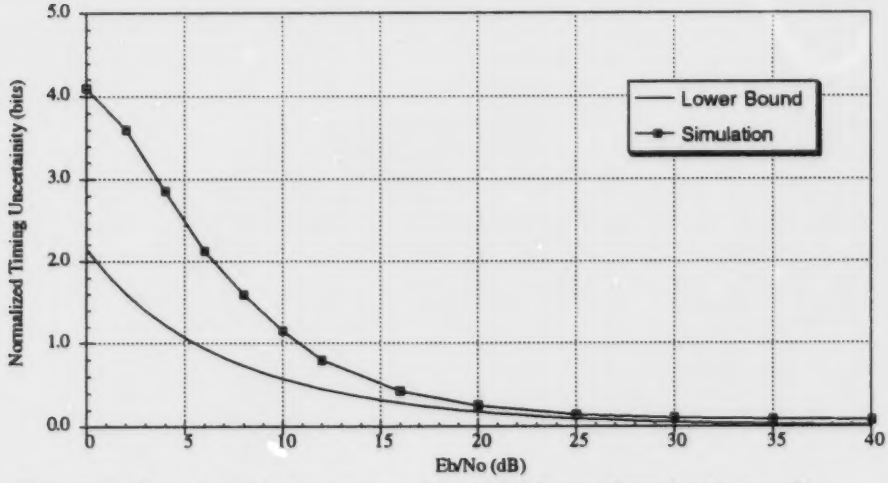

Figure 7 Coarse timing accuracy for a 28-bit synchronization code.

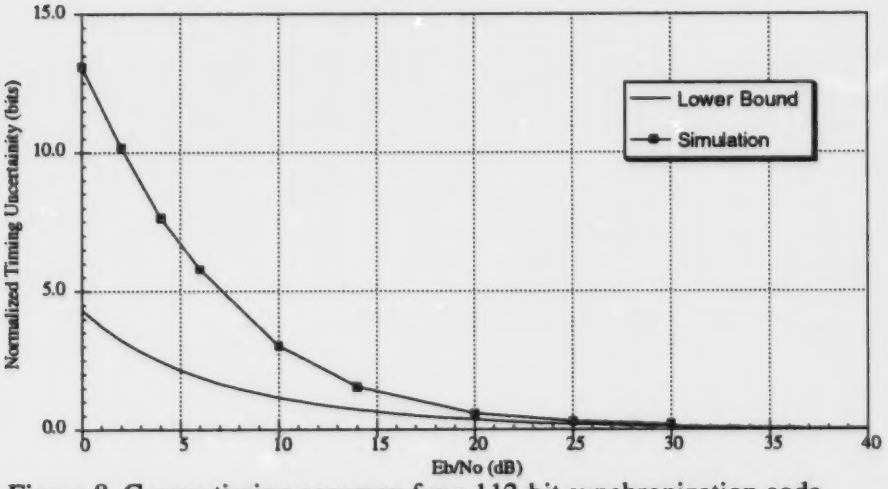

Figure 8 Coarse timing accuracy for a 112-bit synchronization code.

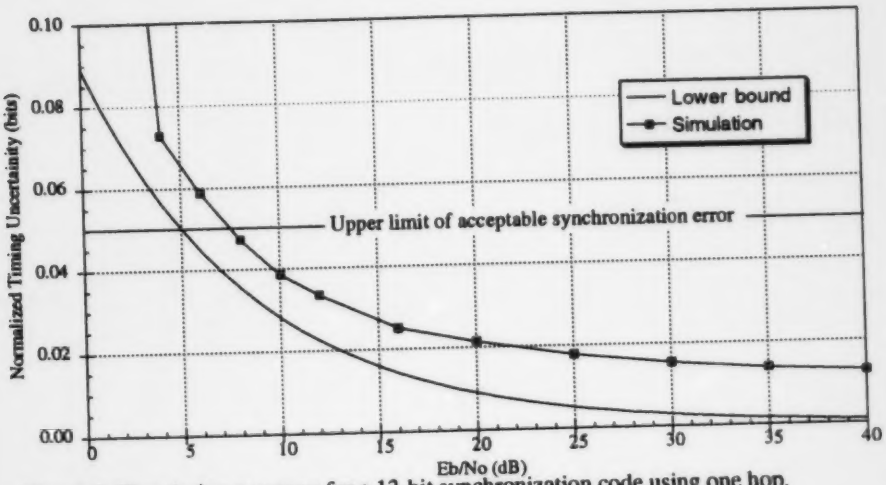

Figure 9 Fine timing accuracy for a 13-bit synchronization code using one hop.

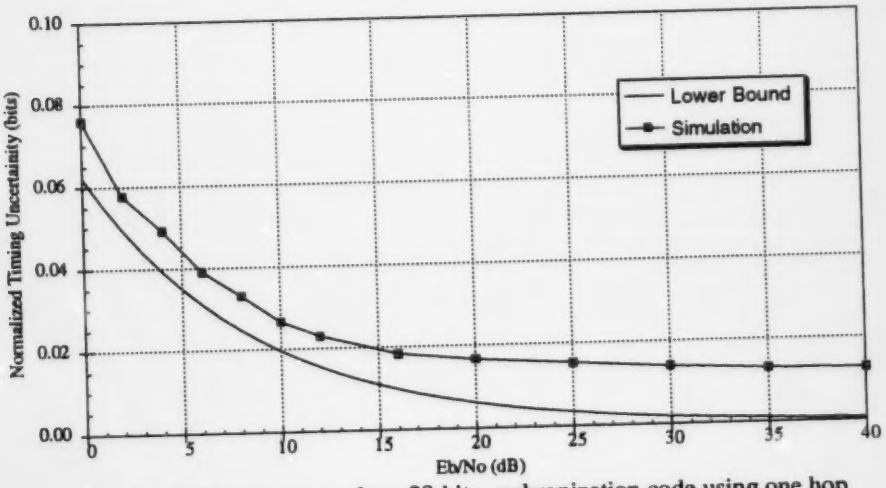

Figure 10 Fine timing accuracy for a 28-bit synchronization code using one hop.

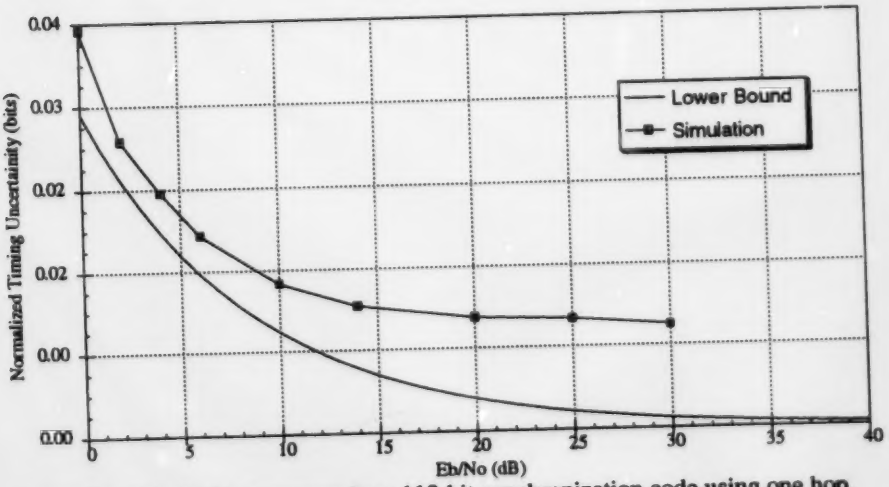

Figure 11 Fine timing accuracy for a 112-bit synchronization code using one hop.

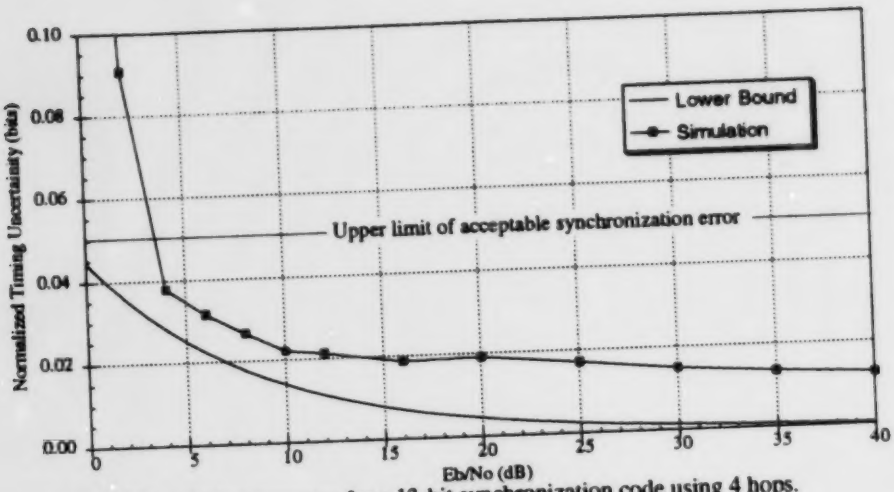

Figure 12 Fine timing accuracy for a 13-bit synchronization code using 4 hops.

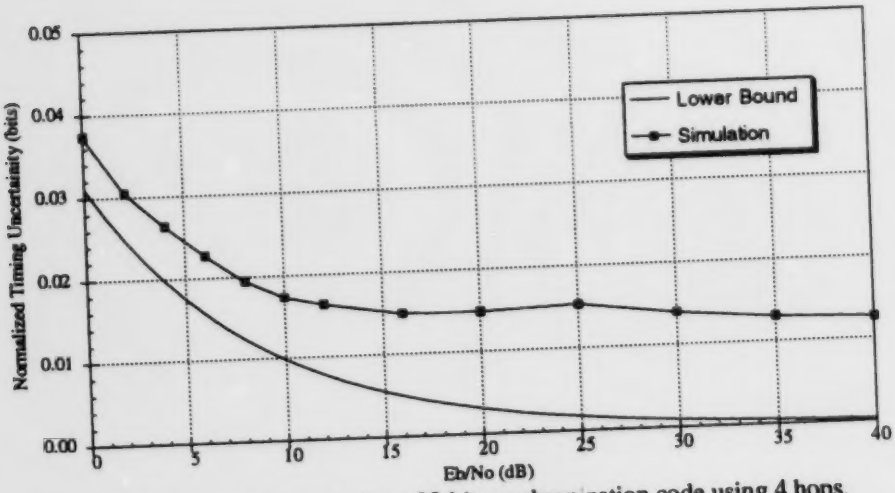

Figure 13 Fine timing accuracy for a 28-bit synchronization code using 4 hops.

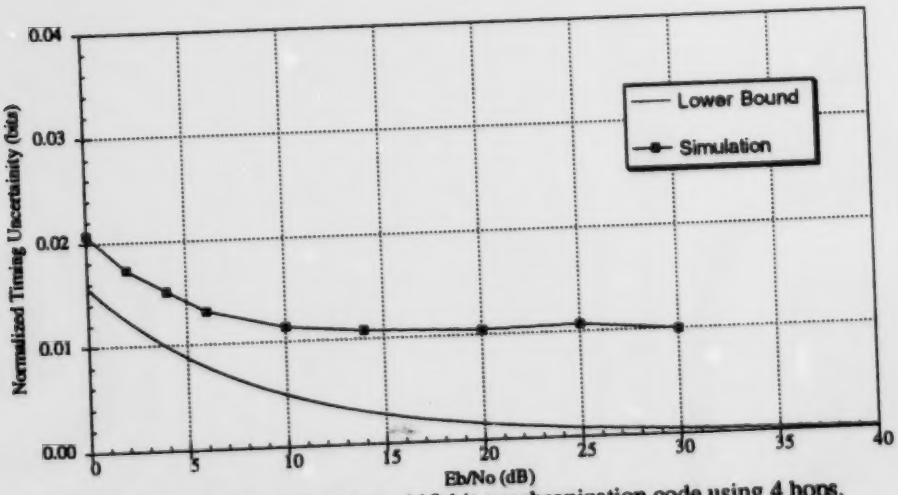

Figure 14 Fine timing accuracy for a 112-bit synchronization code using 4 hops.

As in the simulation of coarse time estimate, the fine time estimation algorithm was shown to function at low values of E_b/N_o. In fact the fine algorithm worked to lower values of E_b/N_o than the coarse algorithm. Figures 9 and 12 (code length = 13) show the upper limit of where the synchronization error is considered to be acceptable, less than 1/20 th of a bit. Using a single hop for the fine time estimate, the algorithm showed that acceptable timing estimates could be achieved down to an E_b/N_o of 8 dB. If a 4-hop average is used, the algorithm was acceptable down to an E_b/N_o of 4 dB.

The improvement of the 4-hop estimate over the 1-hop estimate approaches the theoretical 3 dB at low E/N_o, but reduces to 0 dB as the E_b/N_o increases (see Figure 15). The error in timing estimates is mainly due to two factors; the error caused by the approximation of the cross correlation function as a triangle and the error caused by variations of the cross correlation value due to noise. At high E_b/N_o the main error source is due to the approximation of the correlation function, which is the same for the 1-hop and 4-hop estimate. Therefore no improvement is possible by averaging. At low E_b/N_o the main source of error is due to noise and the theoretical 3 dB improvement can be realized by using the 4-hop average (assuming the burst-to-burst timing offset is random). It should be noted that the error due to a poor approximation of the autocorrelation function is less than $0.02t_b$ for all cases investigated.

Frequency Offset Estimates

Simulations to determine the accuracy of the frequency offset calculations were done for the same code lengths as above (13 bits, 28 bits and 112 bits). Frequency errors of 3%, 6%, 9%, and 12% of the hop rate were introduced into the simulation. Figure 16 shows the resultant RMS error between the actual frequency offset and the calculated offset for a code length of 13 bits. The RMS error is expressed as a ratio of the hop rate. Figure 16 shows the RMS error using only 1 hop in the calculation, Figure 17 shows the RMS error using a 4-hop average in the calculation. For E_b/N_o values greater then 5dB the errors for the various frequency offsets are closely grouped around the average.

The average RMS errors versus E_b/N_o are given in Figure 18 for all 3 code lengths. These errors are compared to the CR bound derived from equation 11. Figure 18 shows that the RMS error for the 4-hop average is 3 dB[1] better than using the 1-hop value for all cases as would be expected. For RMS errors of less then 3% the bit error rate (BER) is degraded by a negligible amount [2]. Using this criteria the estimate is sufficiently accurate down to an E_b/N_o of 24 dB using the 1-hop value and accurate down to an E_b/N_o of 18 dB the 4-hop value. Table 4 shows the accuracy cutoff value for the various code lengths in terms of E_b/N_0. If necessary the estimate could be improved by using more hops in the frequency offset estimate.

---

[1]The increase of accuracy at high E_b/N_o is defined to be $1/\sqrt{n}$, where for our case n = 4.

[2]Analysis of an imperfect carrier reference for a BPSK signal in [5] shows that a 3% carrier frequency error has a minimal effect on the bit error rate.

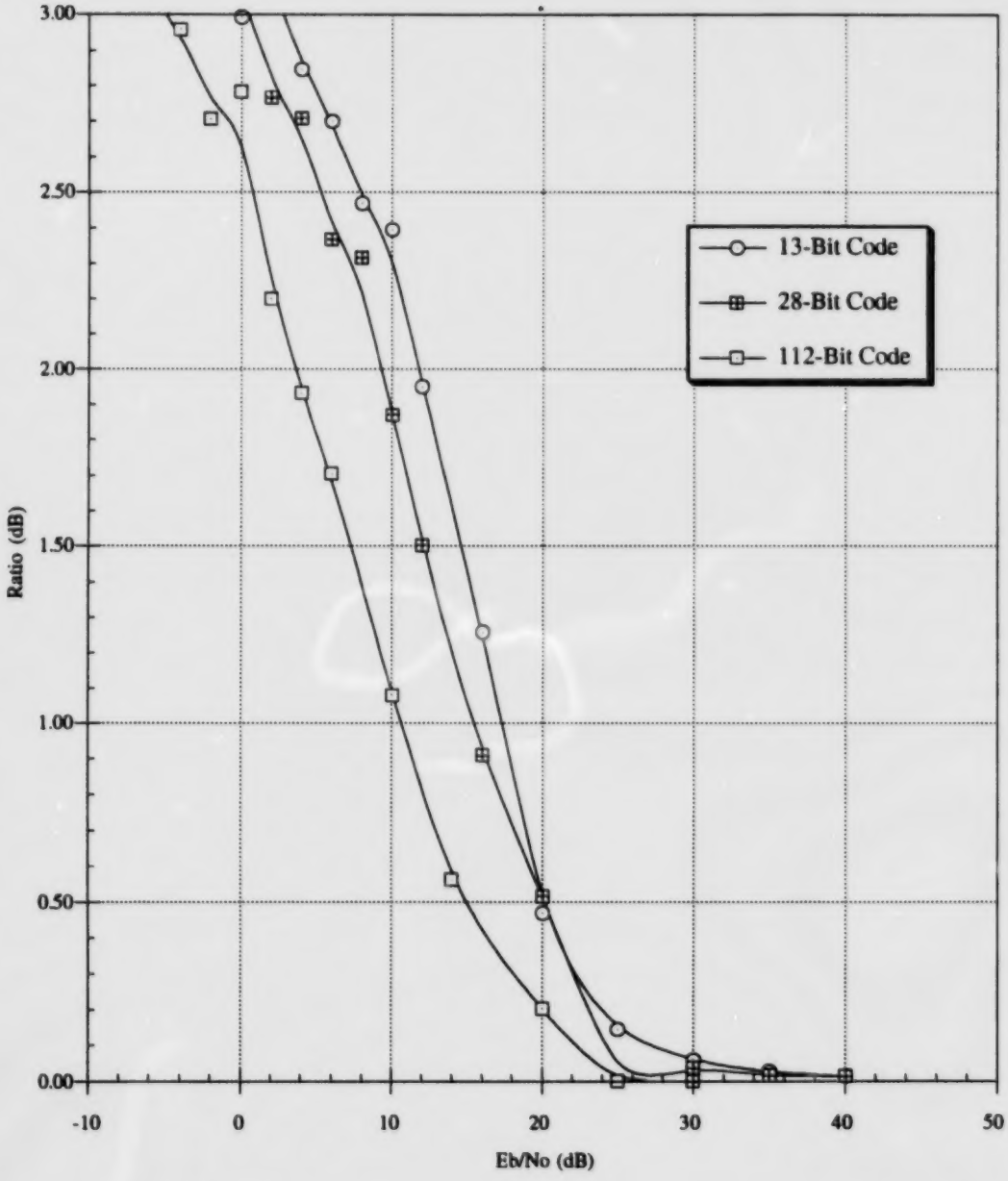

Figure 15 Ratio of 4-hop fine timing estimate over 1-hop fine timing estimate.

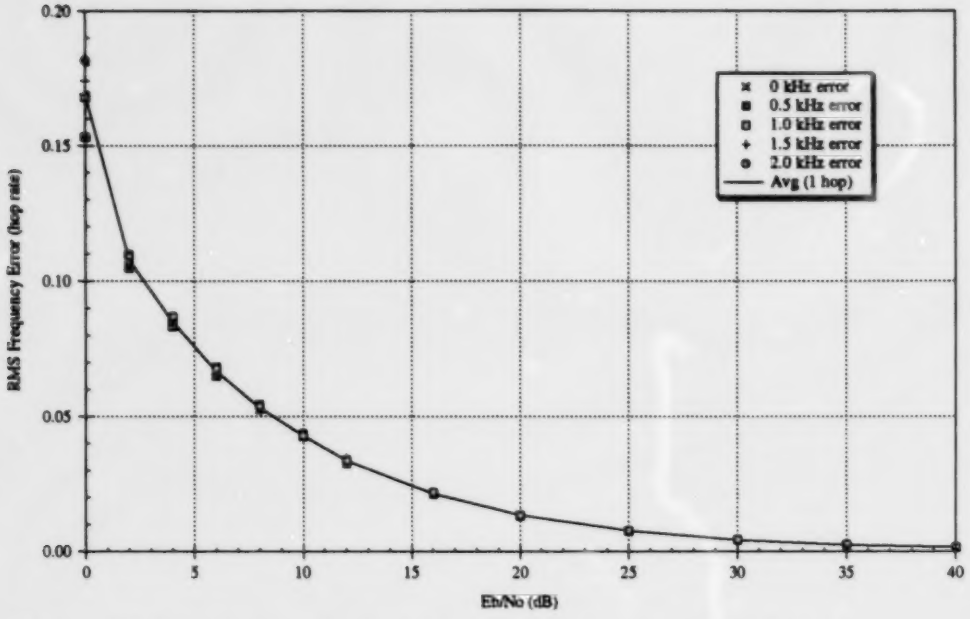

Figure 16 Frequency offset calculation accuracy using one hop.

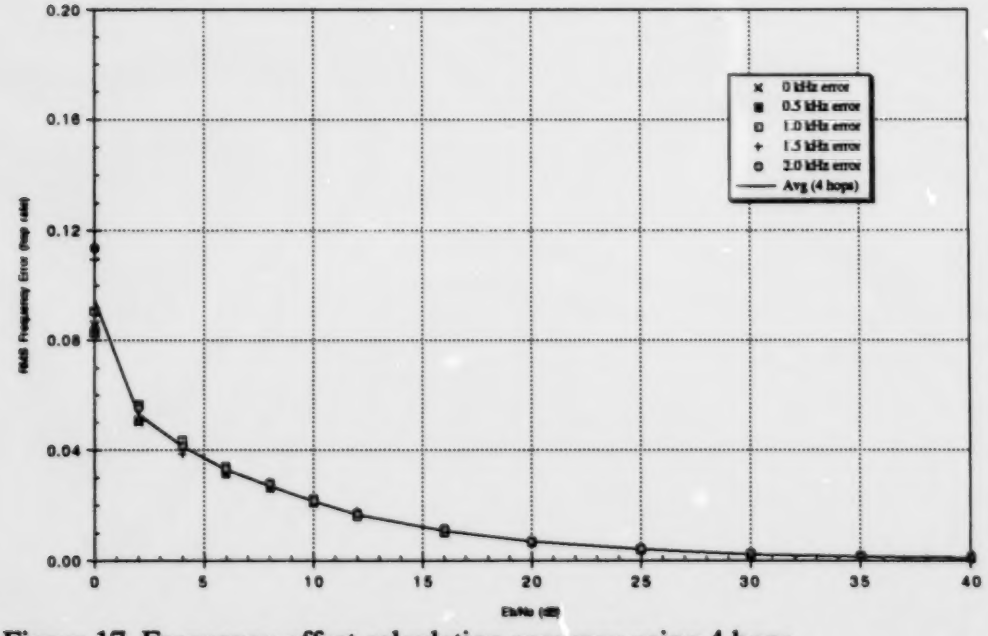

Figure 17 Frequency offset calculation accuracy using 4 hops.

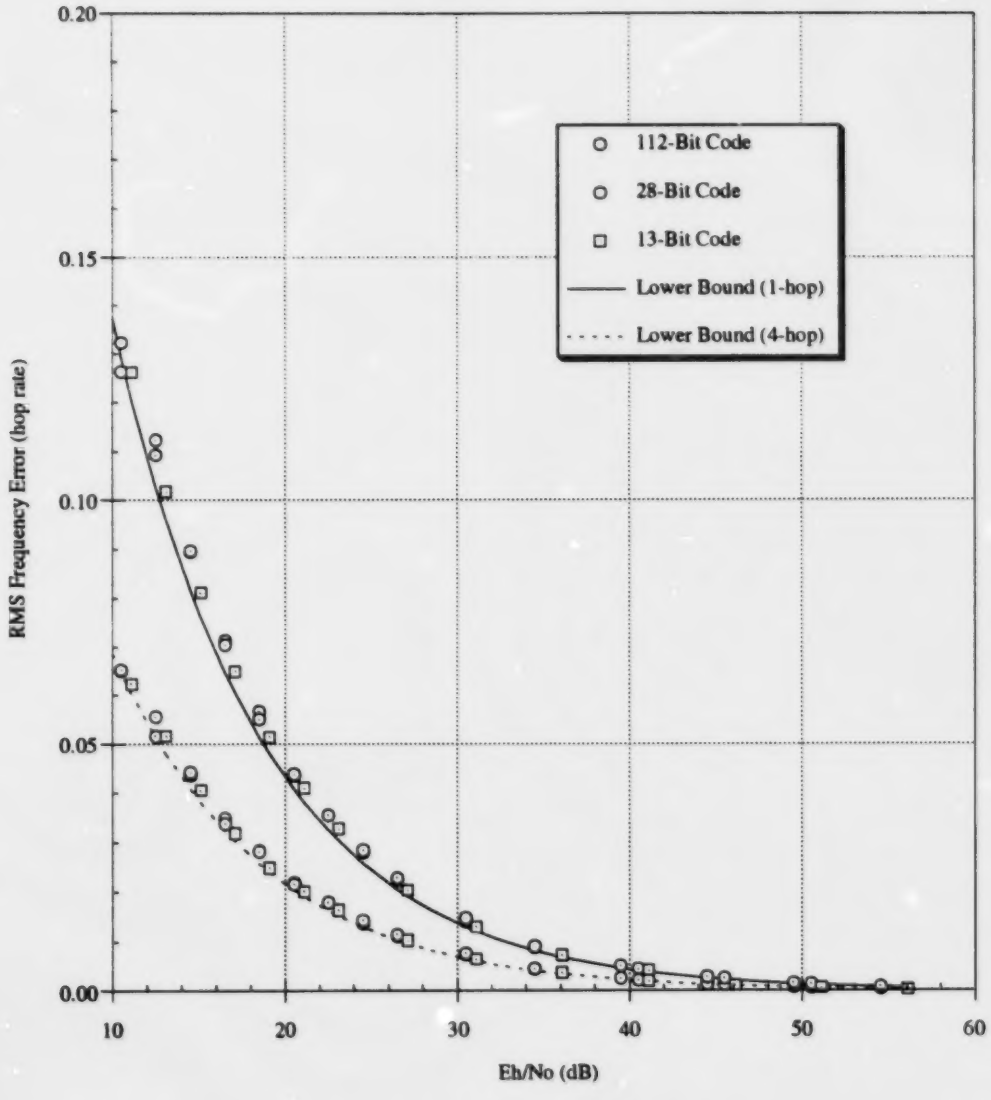

Figure 18 Frequency offset determination accuracy.

	E_b/N_o	E_b/N_o		
		13 Bit Code	28 Bit Code	112 Bit Code
1 Hop Estimate	24dB	13dB	10dB	4dB
4-Hop Estimate	18dB	7dB	4dB	-2dB

Table 4 Accuracy cutoff value for 3 code lengths.

The effects of temperal and frequency errors are graphically depicted in the ambiguity diagrams for the three synchronization codes (Figures 19, 20, and 21) at high E_b/N_o. As can be seen in the figures the sidelobes increase as frequency offset increases. As these sidelobes increase, the probability that the synchronization algorithm will choose the wrong offset also increases. Comparing the three figures, it is apparent that the longer the synchronization code, the better its performance is against frequency offset errors.

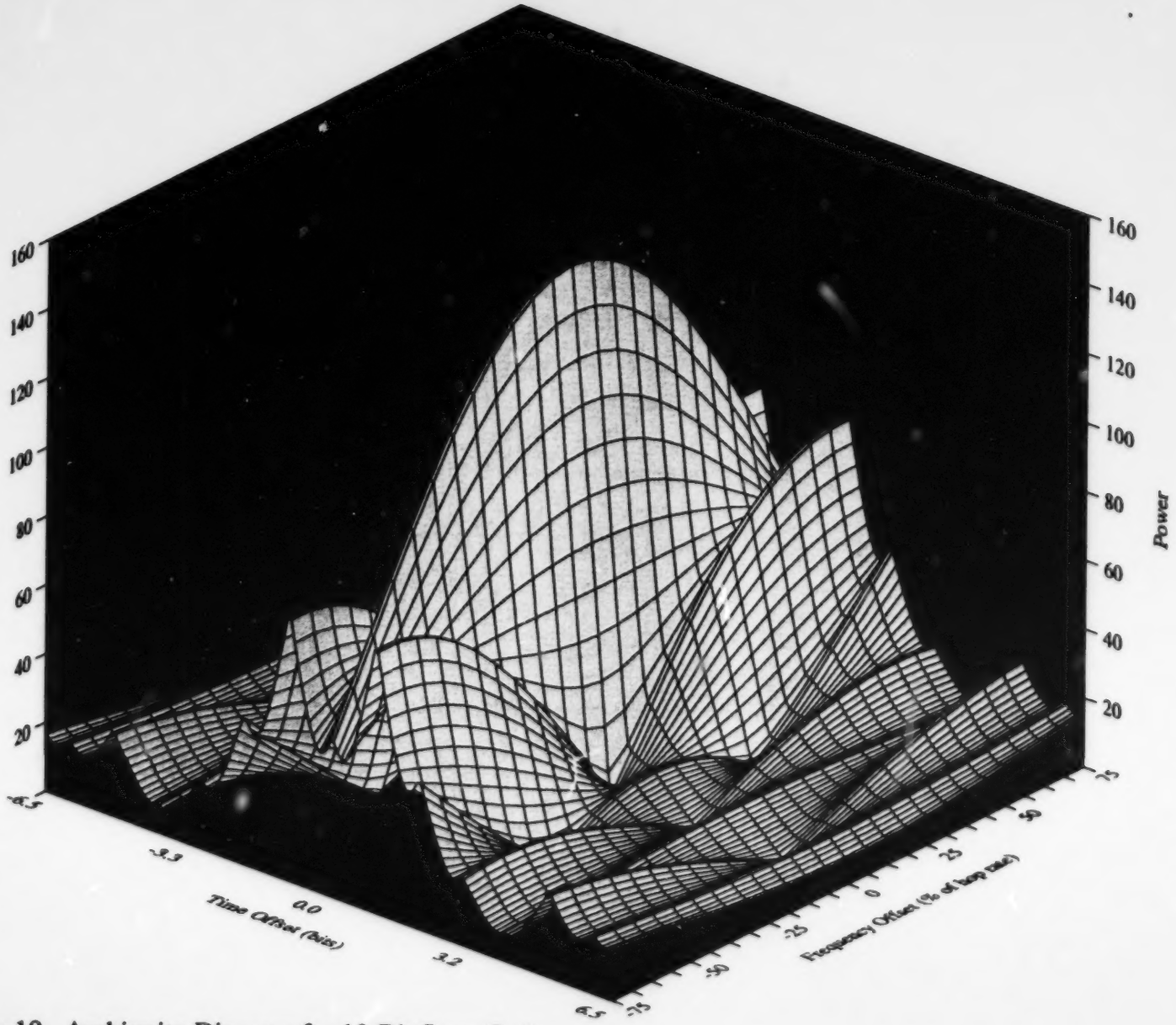

Figure 19. Ambiguity Diagram for 13-Bit Sync Code

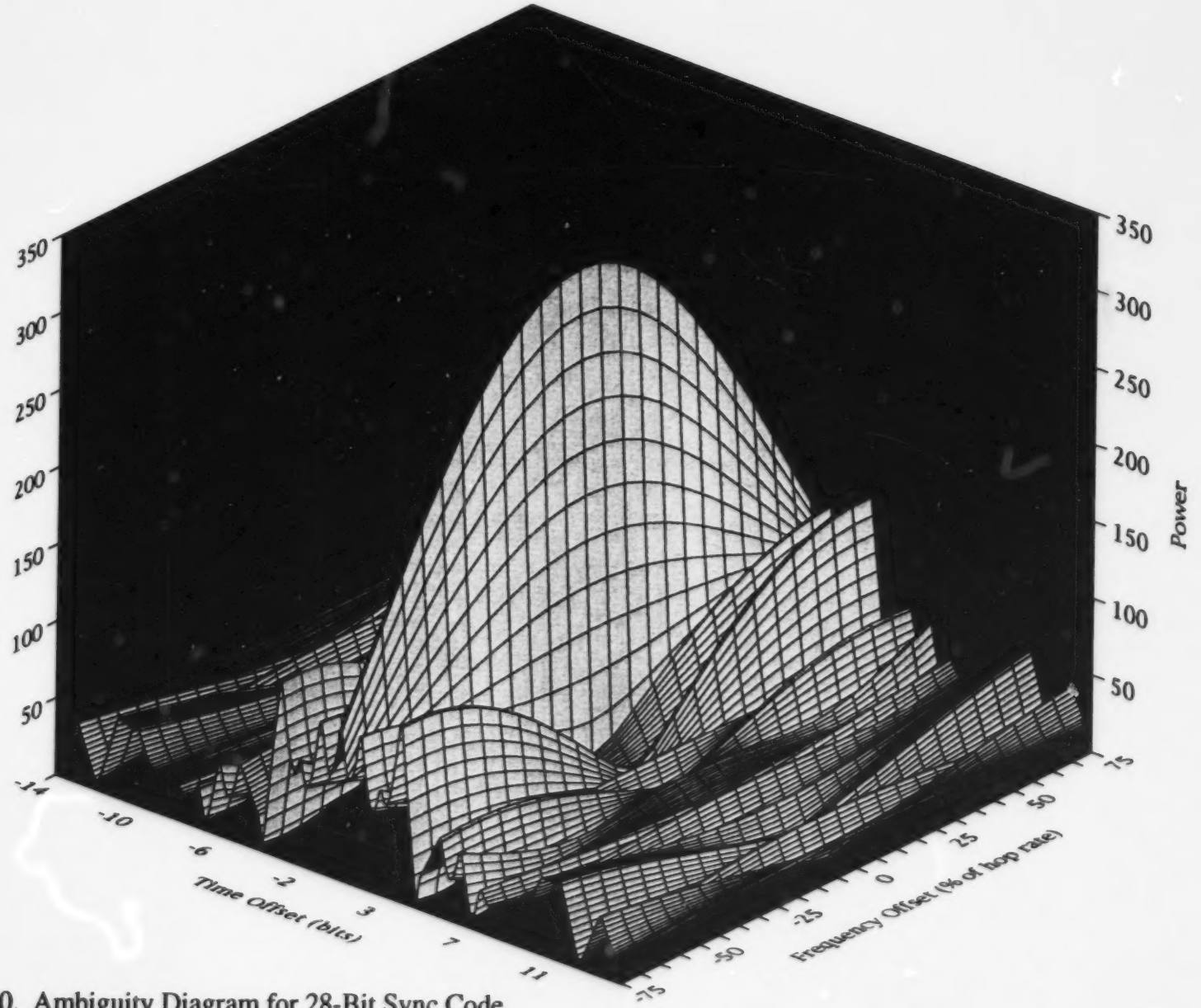

Figure 20. Ambiguity Diagram for 28-Bit Sync Code

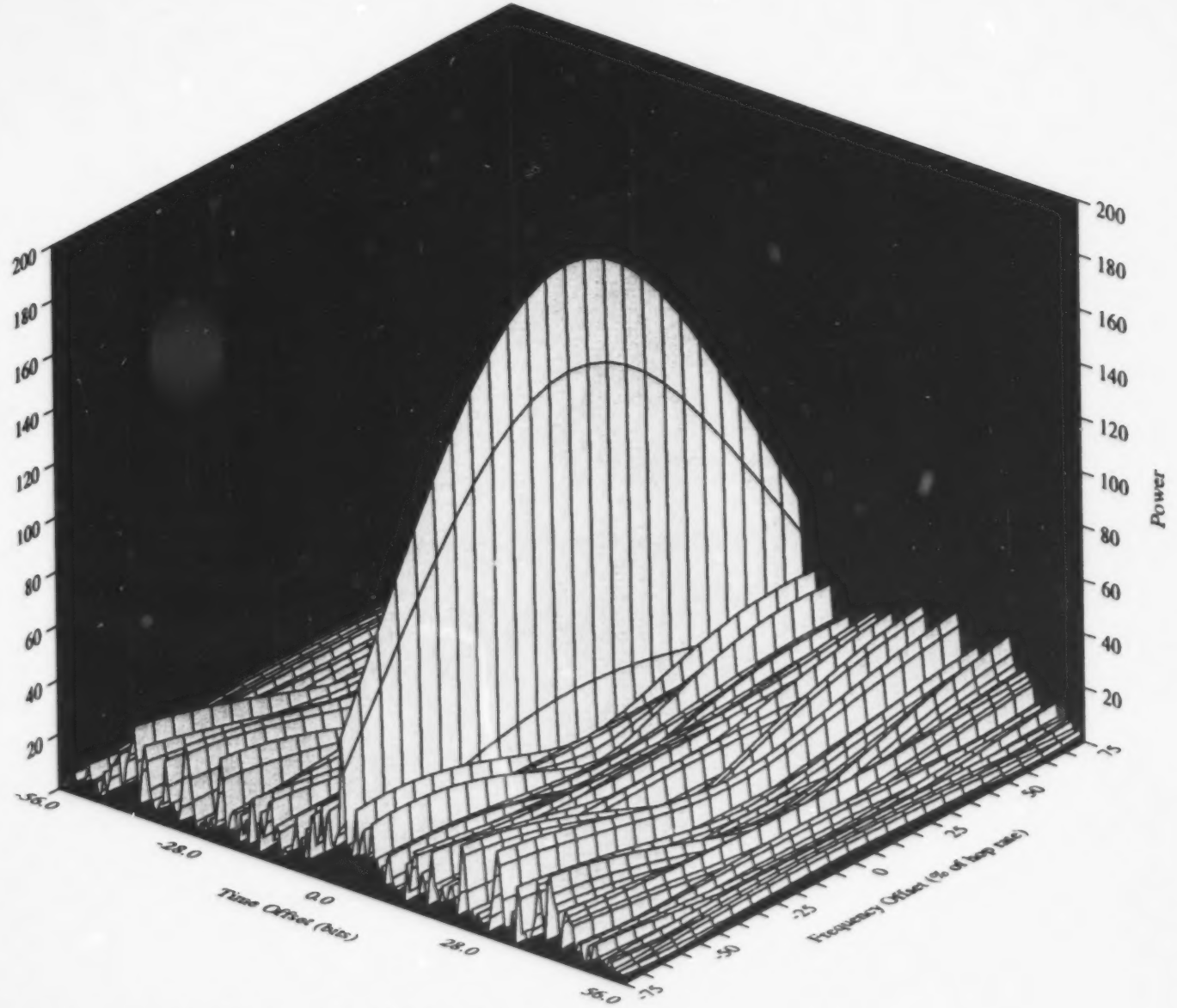

Figure 21. Ambiguity Diagram for 112-Bit Sync Code

5.0

CONCLUSION

This document has described an method for downlink synchronization that is at least as robust as the spread spectrum communications system. The associated algorithm was used with synchronization codes of three different lengths: a short code (13 bits), a medium length code (28 bits), and a long code (112 bits). The selection of which code to be used would be dependant upon the required accuracy of the downlink time and frequency and the data link waveform structure. In general, the longer the code, the greater the accuracy.

References

1. W.R. Seed; "Downlink Synchronization of Frequency-Hopped DPSK"; TTCP STP-6 Proceedings of the 18th Meeting, Vol. III-A, pp. 61-68; January 1991; Ottawa

2. M.I. Skolnik; "Introduction to Radar Systems"; McGraw-Hill; New York; 1980

3. W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling; "Numerical Recipes, the Art of Scientific Computing"; Cambridge University Press, pp. 202-203

4. W. R. Seed; "Square-law Device Analysis for the Detection and Timing Estimation of a Burst BPSK Signal"; DREO Technical Note 91-6; June 1991; Ottawa

5. J.K. Holmes; "Coherent Spread Spectrum Systems"; John Wiley & Sons; New York; 1982

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Defence Research Establishment Ottawa
Ottawa, Ontario
K1A 0Z4

2. SECURITY CLASSIFICATION (overall security classification of the document including special warning terms if applicable)

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C or U) in parentheses after the title.)

Simulation of Downlink Synchronization for a Frequency-Hopped SATCOM Systems (U)

4. AUTHORS (Last name, first name, middle initial)

Wagner, Lyle C.

5. DATE OF PUBLICATION (month and year of publication of document)

Apr 1992

6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)

39

6b. NO. OF REFS (total cited in document)

5

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

DREO Technical Note

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

EHF SATCOM, Electronics Division, Defence Research Establishment Ottawa
Ottawa, Ontario, K1A 0Z4

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)

Project D6470

9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written)

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)

DREO Technical Note 92-9

10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor)

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
( ) Distribution limited to defence departments and defence contractors; further distribution only as approved
( ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
( ) Distribution limited to government departments and agencies; further distribution only as approved
( ) Distribution limited to defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). however, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

Unlimited Announcement

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM

13. **ABSTRACT** (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

(U) Many military communication systems are required to operate in environments where an enemy may attempt to interfere with the communication. One of the most serious threats is the use of jammers. This threat has caused designers of communications systems to counter with the use of spread spectrum techniques such as frequency-hopping.

(U) Frequency-hopped spread spectrum satellite communication systems provide robust communications resilient to many types and levels of jamming. Unfortunately this increase in resilience is offset by an increase in complexity while establishing the communication link, termed synchronization. This document describes a downlink synchronization process that involves the transmission of synchronization hops by the satellite and a two-step ground terminal synchonization procedure. In addition computer simulation results are documented which demonstrate that the proposed synchronization algorithm is able to operate in environments as severe as the communication link.

14. **KEYWORDS, DESCRIPTORS or IDENTIFIERS** (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Computer Simulation, Satellite Communication, SATCOM, MILSATCOM, Spread Spectrum, Frequency-Hopped, Synchronization